

Docket No. 100201028
jdawson@koppelpatent.com

**DATA MANAGEMENT SYSTEM AND METHOD
WITH MULTI-PORT PROCESSOR**

BACKGROUND

[0001] Electronic devices which capture, create, store, manipulate or transfer digital music, sound, images, movies or other encoded data have become more prevalent with the advent of less expensive semiconductor processing and increased consumer demand. Applications such as portable MP3 (Moving Picture Experts Group layer 3 standard) players, PDAs (electronic personal data assistant), digital cameras and digital voice recorders continue to gain popularity. The general trend for each of these electronic applications is to provide increased capability and inter-operability with reduced form factor.

[0002] One standard that defines inter-operability between electronic devices includes the On-The-Go Rev 1.0 supplement (OTG) to the USB (universal serial bus) 2.0 standard. The OTG standard enables an MP3 player to connect to another MP3 player to transfer music files, a camera to connect to a printer to print a picture, or a PDA to connect to a cell phone to enable mobile web surfing. See Bethanee Martin, USB On-The-Go Spec Signals Developers To Proceed With a New Generation of Mobile Products Capable of Point-to-Point Data Exchange, ¶5 (December 18, 2001). When one OTG-enabled device (the "A-device") is attached to another (the "B-device"), a pull-

up resistor within the B-device raises one line in their connection by a predetermined voltage amount. The A-device senses the voltage change and responds by sending an initialization signal. The B-device responds with bus bandwidth, access frequency, latency and error handling behavior requirements. The A-device assigns the B-device a unique identification number for data addressing. See Michael Gowan, How it Works: USB, PCWorld.com, ¶6 (December 30, 1999). In subsequent communications, data is divided into 64-byte portions, each of which includes both addressing information and the data itself. Id. OTG cabling is asymmetric, with a Mini-A side and a Mini-B side. Devices have only one Mini-AB receptacle that accepts both Mini-A and Mini-B plugs. USB On-the-Go: A Tutorial, P.6, Koninklijke Philips Electronics (January 2002). The OTG-enabled host can have only one peripheral connected to it at a time, thus limiting its use.

[0003] One example of an electronic device which includes USB controller functionality is the STMP3300 audio decoder for use with MP3 players. Offered by Sigmatel, Inc., a USB port on the device enables MP3 and WMA (Windows™ media audio) file downloads. William Wong, 1-Chip MP3 Player Includes A USB Controller, Electronic Design, June 4, 2001, V 49, I 12, at 37. Unfortunately, the USB port requires a USB host PC (personal computer) to enable the USB connectivity.

[0004] An improved data management system and method is particularly important, since current standards do not allow serial connection of multiple B-devices with an A-device on either end and without the use of a PC.

SUMMARY

[0005] A data management system and method has a plurality of data ports coupled to a processor. The processor is programmed to test for the presence of a controller alternately through each of the plurality of data ports.

[0006] In an embodiment, the method is described by testing for the presence of a controller through a first port using a processor and testing for the presence of said controller through a second port if said controller is not found through said first port.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The components in the figures are not necessarily to scale, emphasis instead being placed upon illustrating the principals of the invention. Moreover, in the figures, like reference numerals designate corresponding parts throughout the different views.

[0008] Figure 1 is a flow diagram of one embodiment that illustrates the transmittal of a handshake through two ports to establish an application identification ("ID") for a processor.

[0009] Figure 2 is flow diagram of an embodiment that illustrates data flow from a controller module to an application module using the application ID established in Figure 1.

[0010] Figure 3 is a block diagram of an embodiment illustrating an application module used to establish the application ID and capture data having the application ID.

[0011] Figure 4 is an exploded perspective view illustrating a portable electronic device with modular application modules to which the invention is applicable.

5 DETAILED DESCRIPTION

[0012] One exemplary embodiment of the preferred embodiment comprises a portable modular system with detachably connectable controller, memory and application modules. In one version of the system, the controller module can
10 be connected to a number of different application modules that are connected in series, with each application module including a pair of opposed data ports for connection to other modules. In one embodiment, the orientation of each application module is reversible, so that a given
15 data port can be directed either towards or away from the controller module without affecting system operation. This system is the subject of application Serial No. 10/307,034, filed November 27, 2002, "Portable Modular Electronic System".

20 [0013] The data management system described herein, in one embodiment, coordinates data flow between the various applications and the controller. A data addressing protocol is established to enable routing of packet data through the system. The protocol can be appended to the
25 packet data in an addressing portion. The protocol can also precede the packet data to enable routing between the applications and controller. The system includes a plurality of data ports coupled to a processor. The processor is programmed to test for the presence of a con-
30 troller alternately through each of the plurality of data ports to receive an application ID from the controller.

Further, the system enables other application modules to be connected serially with the controller module and the processor in the first application module.

[0014] Figure 1 is a flow diagram, for one embodiment of the invention, which illustrates how a processor in an application module receives an application ID from a controller module. In a packet-data system, the application ID is contained in an addressing portion of a data packet and represents the electronic address of the processor in the system. Receipt of the application ID enables for the processor to later request and retrieve data. The application module includes at least two data ports, either of which can be connected to a controller module, depending upon the relative orientations of the modules. The application module is turned on (block 100) and a controller handshake is transmitted by the processor through a first data port (block 105) of a hub to determine whether a controller module is connected to that port. Incoming data from a second data port in the hub is inhibited to avoid interference with the controller handshake transmission (block 110). The protocol for the handshake transmission includes information that enables the controller module to respond to the controller handshake with an acknowledgement if the controller module is connected to the first port. For example, the controller handshake could include a request to inhibit data transmission from the controller module and send an acknowledgement that the controller handshake was received. Or, the controller handshake could include other identifying information such as protocol version number, application module type, or an authentication certificate that en-

ables the controller module to respond with an acknowledgement.

[0015] If an acknowledgement is sent to the processor (block 115) within a pre-determined time period, the processor transmits an application ID request to the controller module (block 120). The controller module responds by transmitting an application ID to the processor (block 125). The application ID is stored in internal memory for later retrieval and use. The controller module also has an ID, and its ID is transmitted to the application module either with the acknowledgement, with the application ID, or subsequently to allow routing of subsequent data requests without having to repeat the controller-handshake process. Data pass-through is then resumed from the first and second ports (block 130) and the handshaking process is complete. The controller appends the application ID onto subsequent data transmitted to the application module to enable proper data routing.

[0016] If the original controller handshake transmittal through the first data port does not result in an acknowledgement (blocks 105, 115), indicating that no controller module is connected to the first port, the application module looks for the presence of a controller module at its second data port by transmitting a controller handshake through the second port (block 235). Data pass-through is inhibited through the first data port (block 140) to prevent interference. If an acknowledgement is returned (block 145), indicating the presence of a controller module at the second port, the processor transmits an ID, request to retrieve a processor ID and resumes data pass-through from both ports (blocks 120, 125,

130). Although the description of Figure 3 includes only two data ports, its process may be extended to more than two data ports. If more than two data ports are provided, each data port would be tested in turn to determine whether a controller module is connected to that data port. The processor would inhibit data pass-through through each data port not being tested.

[0017] Figure 2 is a flow diagram that shows a possible scheme for data retrieval and capture by the application module using the processor ID acquired in Figure 2. The processor in the application module transmits a data request to the controller module (block 200) through the hub. The controller module sends a request for the data to memory (block 210) and transmits the retrieved data with the processor ID to the application module (block 215). The application module monitors the incoming data for data having its processor ID (block 220) to route the incoming data to the processor. The retrieved data is captured and provided to the processor for processing (block 225). If more data is needed by the processor, a new data request is transmitted using the targeted device's ID for routing (block 230).

[0018] Figure 3 illustrates an implementation of an application module 300. It includes a data bus 310 in communication with a user interface 320, an internal memory 325, a processor with an embedded controller 330, and a hub 335. The hub 335 can communicate with the controller module 340 and a second application module 345 through first and second ports in the hub, respectively. The controller module 340 is connected to redundant memories A and B for data storage and access.

[0019] The processor 330 implements an application such as an MP3 player, PDA, digital camera or digital voice recorder. The processor 330 accepts user input and provides status information to the user through a user interface 320. The processor 330 is programmed to transmit a controller handshake signal through one data port and to inhibit data pass-through at the second port, and to then transmit a controller-handshake signal through the second data port and inhibit data pass-through at the first data port in case no controller module is located at the first port, to establish the initial communication with the controller module 340. Switching in the hub for the initial communication is controlled by the processor. The controller module 340 is programmed to return an application module ID to the processor 330 in response to receiving the controller-handshake signal. As described above, the application module's internal memory 325 stores the application module ID for use by the hub and processor. The internal memory 325 also enables efficient buffering of data retrieved from memory A and B for use by the processor 330.

[0020] The hub 335 can be either a switching hub or a router having at least one switch that forwards the data to the appropriate data port or to the processor 330, based on the ID contained in the addressing portion of the data packet (see Figure 1). For example, if data entering through the first data port does not match the application ID, the processor is programmed to cause the hub to switch so that the data continues through the second port. If the data came from the data port in communication with the controller module 340, the data would be

communicated to the second application module at the opposite data port (if attached). Data retrieved from memories A or B for the processor 330 would be addressed with the application ID and switched by the hub 335 to the processor 330.

[0021] The internal memory 325 may be integrated onto a single chip with the processor 330. The data bus 310 is illustrated with electrically conductive paths between the processor 330, user interface 320, and internal memory 325. Other signal transport mechanisms, such as an optical bus, may also be used. A wireless scheme utilizing BluetoothTM or other wireless technology could also be provided for the data path between the controller module 340, memories A and B and the application module 300.

[0022] The invention has numerous embodiments, such as a portable handheld modular consumer electronic product such as that shown in Figure 4. Such a modular system allows a controller module, such as that illustrated in Figure 3, and multiple memories to be packaged together by the consumer with whichever application modules the consumer desires. In Figure 4, the controller module 340 is shown aligned for electrical and mechanical connection with the application module 300 through an electrical connector 400 in the application module and a complementary, opposed connector (not shown) in the controller module 340, and mechanical connectors 410. The controller module 340 manages files sent from the application module 300 to memories A and B. As shown in Figure 3, the application module 300 can be connected in turn to a second application module 345 through additional electrical and mechanical connectors (400, 410). In such a case, the

controller module 340 may distinguish between data from the different application modules by the data-addressing scheme described in connection with Figures 1 through 3.

[0023] Memories A and B are shown aligned for electrical connection with the controller module through a pair of electrical connectors 400 in the controller module, and a complementary pair of electrical connectors (not shown) in the memories, one for each memory. Each memory can be individually replaced if it goes bad, and a new memory installed with either the same or an 180°-rotated orientation with respect to the controller module 340.

[0024] While various embodiments of the invention have been described, it will be apparent to those of ordinary skill in the art that many more embodiments and implementations are possible within the scope of this invention. For example, the application module could have more than two ports for connection to other modules, in which case the processor could be programmed to transmit a controller handshake through each data port in succession until a controller is located. Accordingly, it is intended that the invention be limited only in terms of the appended claims.